FIG. 1

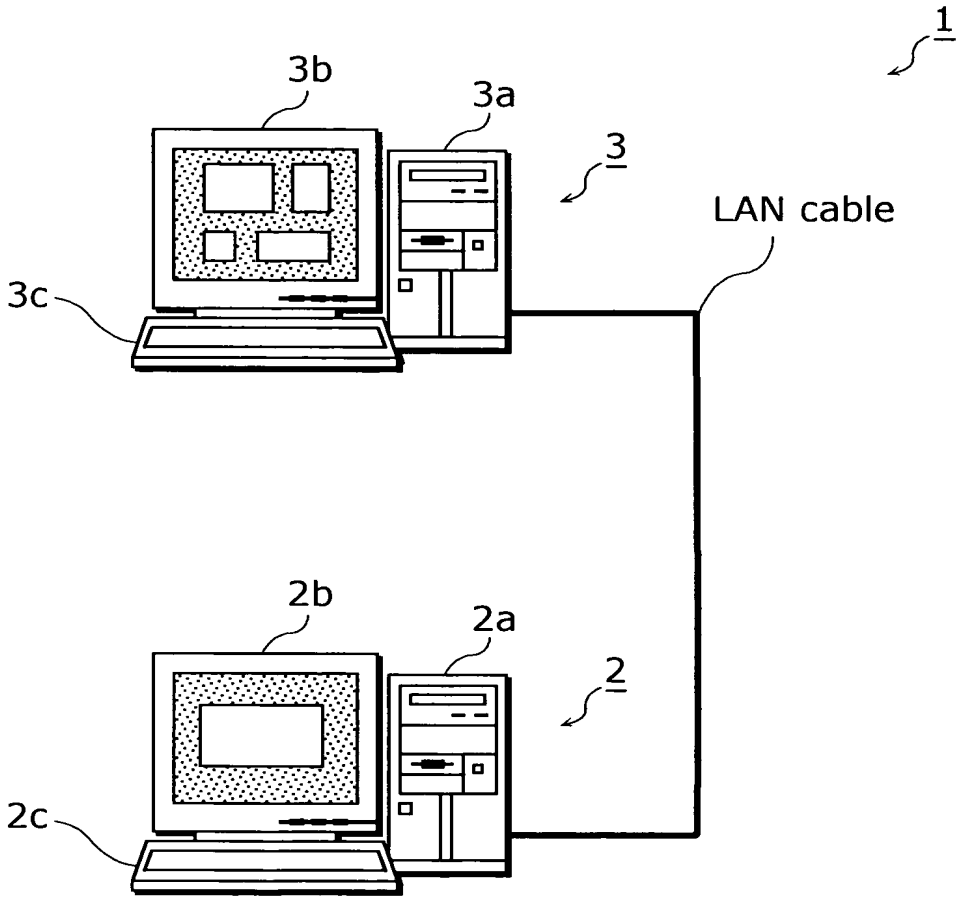


FIG. 2

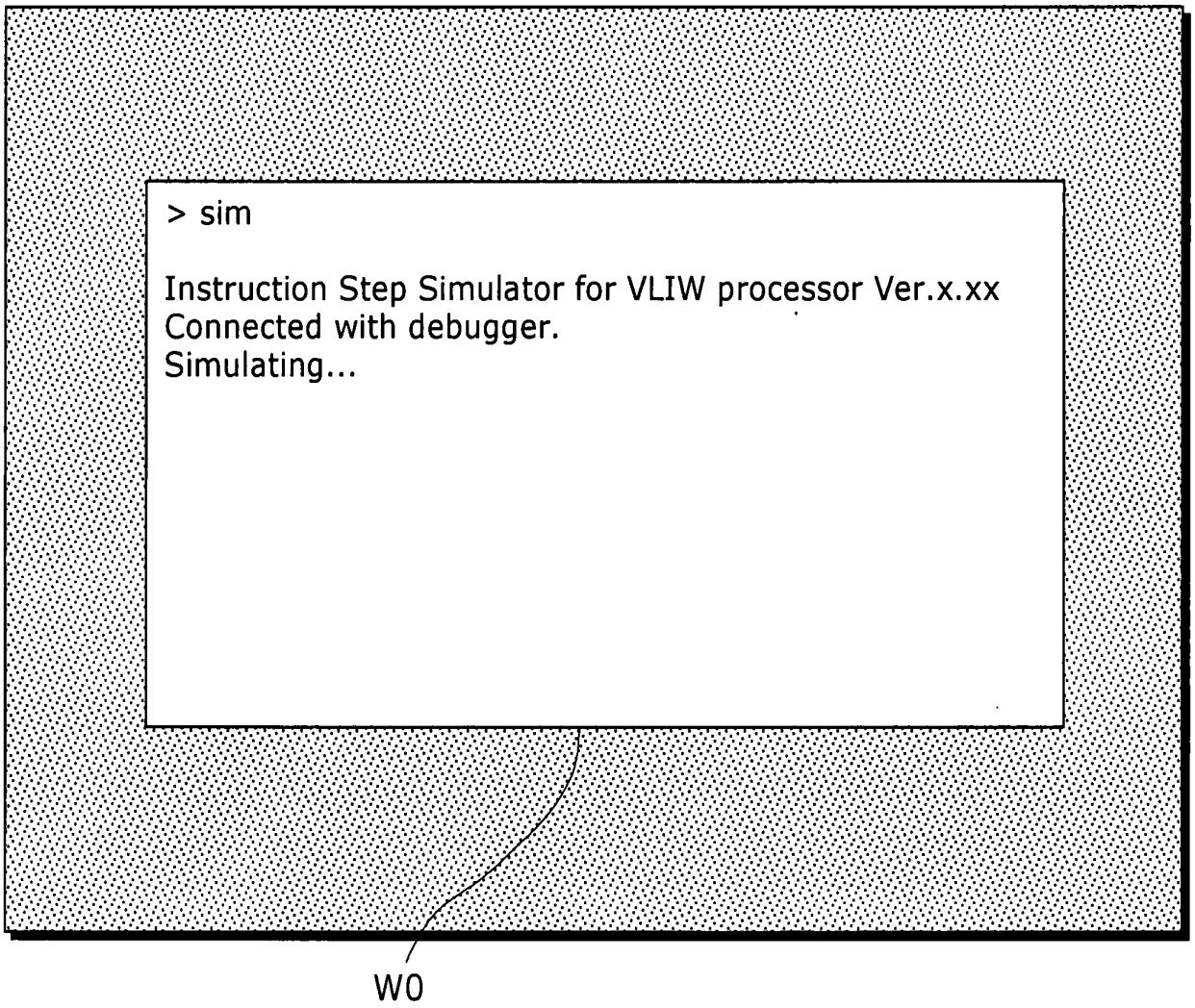
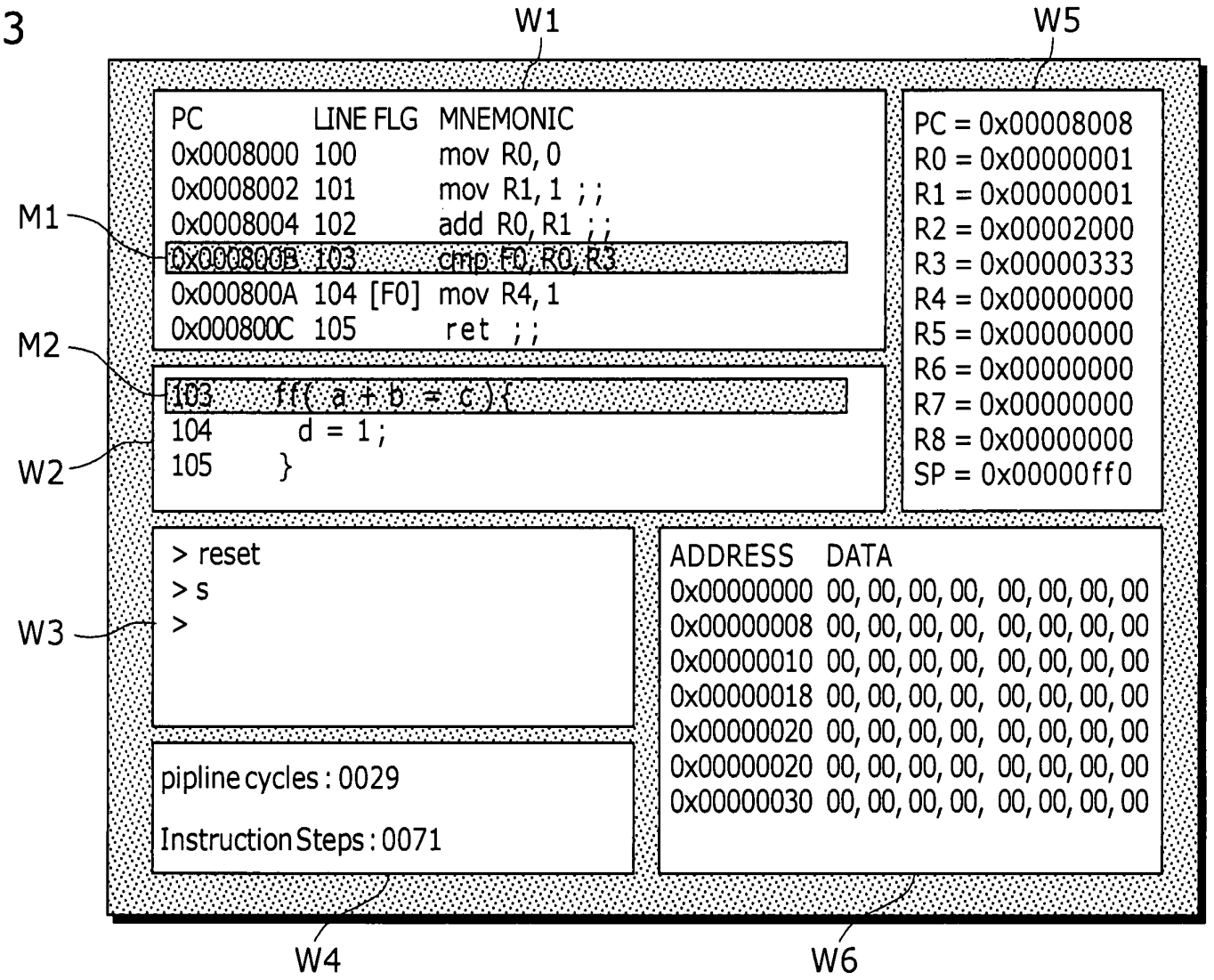

> sim
Instruction Step Simulator for VLIW processor Ver.x.xx
Connected with debugger.
Simulating...
W0

FIG. 3


W1
W5
M1
M2
W2
W3
W4
W6
PC LINE FLG MNEMONIC
0x0008000 100 mov R0, 0
0x0008002 101 mov R1, 1 ; ;
0x0008004 102 add R0, R1 ; ;
0x000800B 103 cmp F0, R0, R3
0x000800A 104 [F0] mov R4, 1
0x000800C 105 ret ; ;
103 ff( a + b = c ){
104 d = 1;
105 }
PC = 0x00008008
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002000
R3 = 0x00000333
R4 = 0x00000000
R5 = 0x00000000
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00000000
SP = 0x00000ff0
> reset
> s
>
ADDRESS DATA
0x00000000 00, 00, 00, 00, 00, 00, 00, 00
0x00000008 00, 00, 00, 00, 00, 00, 00, 00
0x00000010 00, 00, 00, 00, 00, 00, 00, 00
0x00000018 00, 00, 00, 00, 00, 00, 00, 00
0x00000020 00, 00, 00, 00, 00, 00, 00, 00
0x00000020 00, 00, 00, 00, 00, 00, 00, 00
0x00000030 00, 00, 00, 00, 00, 00, 00, 00
pipline cycles : 0029
Instruction Steps : 0071

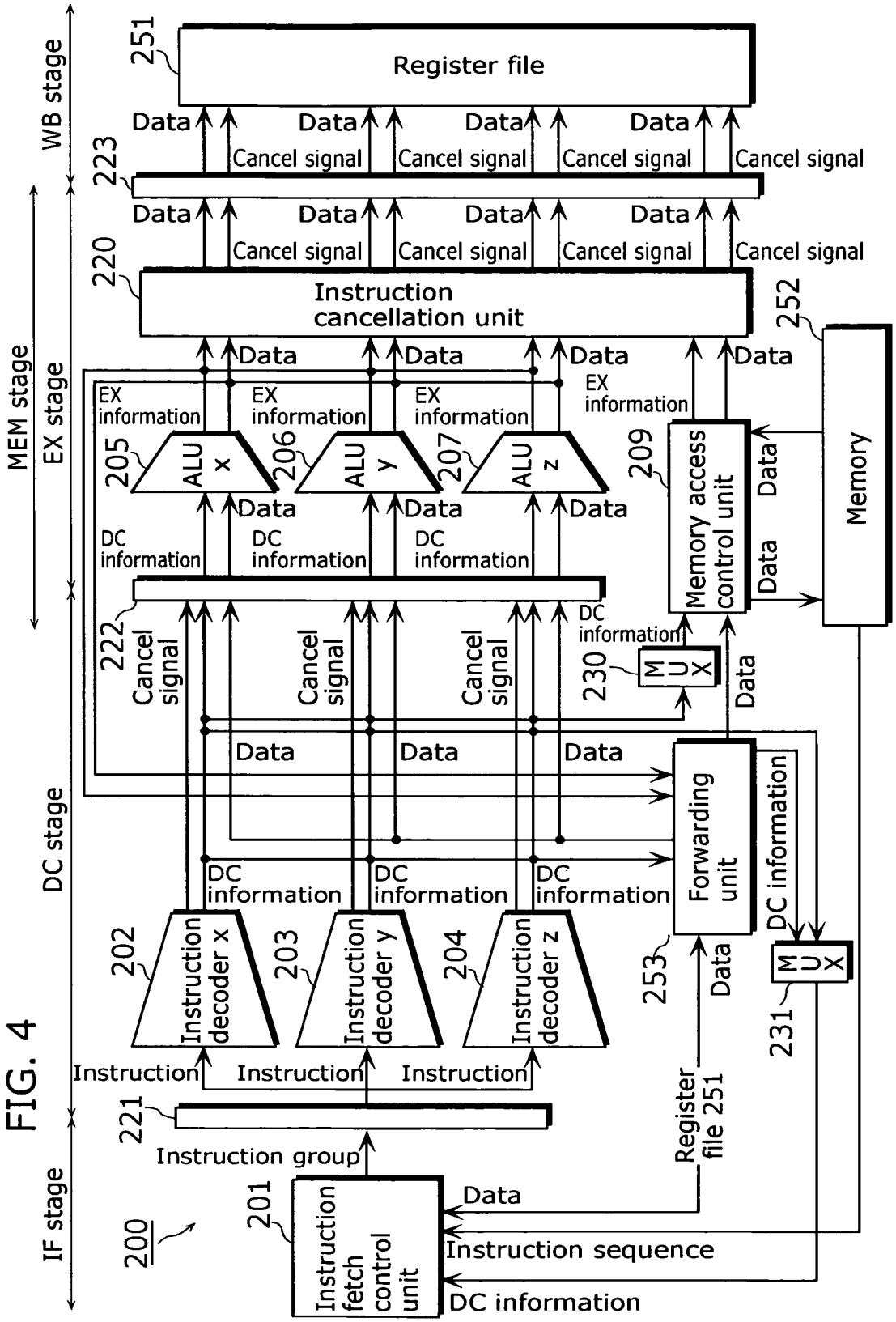

WB stage
MEM stage
EX stage
DC stage
IF stage
251
Register file
Data
Cancel signal
223
220
Instruction cancellation unit
252
Memory
EX information
205
ALU x
206
ALU y
207
ALU z
209
Memory access control unit
DC information
222
230
MUX
231
Forwarding unit
253
202
Instruction decoder x
203
Instruction decoder y
204
Instruction decoder z
Instruction
221
Instruction group
200
201
Instruction fetch control unit
Register file 251
Instruction sequence

FIG. 4

FIG. 5

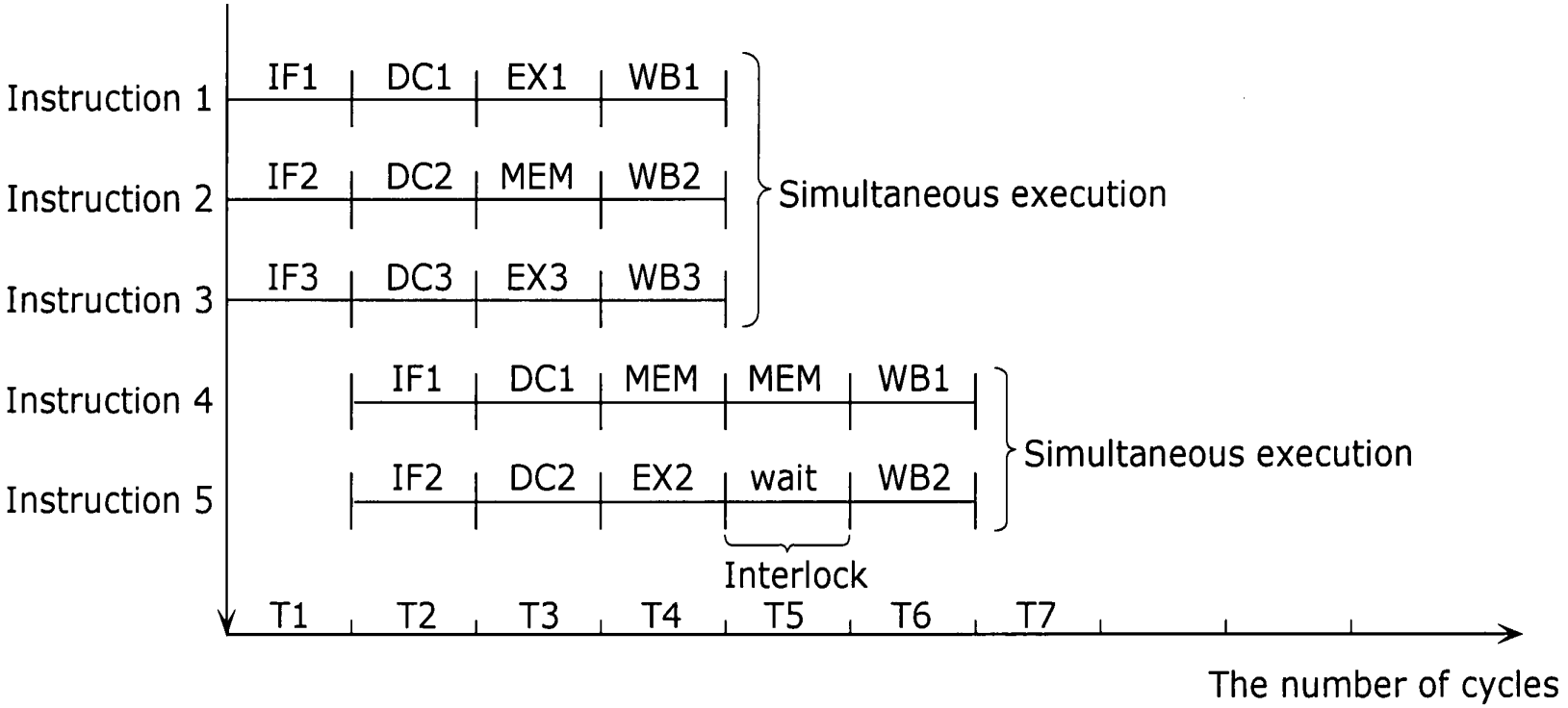

Instruction 1
IF1
DC1
EX1
WB1
Instruction 2
IF2
DC2
MEM
WB2
Simultaneous execution
Instruction 3
IF3
DC3
EX3
WB3
Instruction 4
IF1
DC1
MEM
MEM
WB1
Instruction 5
IF2
DC2
EX2
wait
WB2
Simultaneous execution
Interlock
T1
T2
T3
T4
T5
T6
T7
The number of cycles

FIG. 6

| | | Address | Mnemonic | Instruction execution contents | Update resource |
|---|---|---|---|---|---|
| Instruction group 1 | Instruction 1 | 0x8000 | sub R0,R1 | R0=R0 - R1 | R0 |
| | Instruction 2 | 0x8002 | add R2,1 | R2=R2 + 1 | R2 |
| | Instruction 3 | 0x8004 | ld R3,(R4+) | R3=mem(R4),R4=R4+4 | R3,R4 |
| Instruction group 2 | Instruction 4 | 0x8006 | st (R4+),R2 | mem(R4)=R2,R4=R4+4 | mem(R4),R4 |
| | Instruction 5 | 0x8008 | or R5,R6 | R5=R5 \| R6 | R5 |

FIG. 7

| Cycle | Update resource |
|---|---|
| N+1 | R0,R2,R3,R4 |
| N+2 | <None> |
| N+3 | mem(R4),R4,R5 |

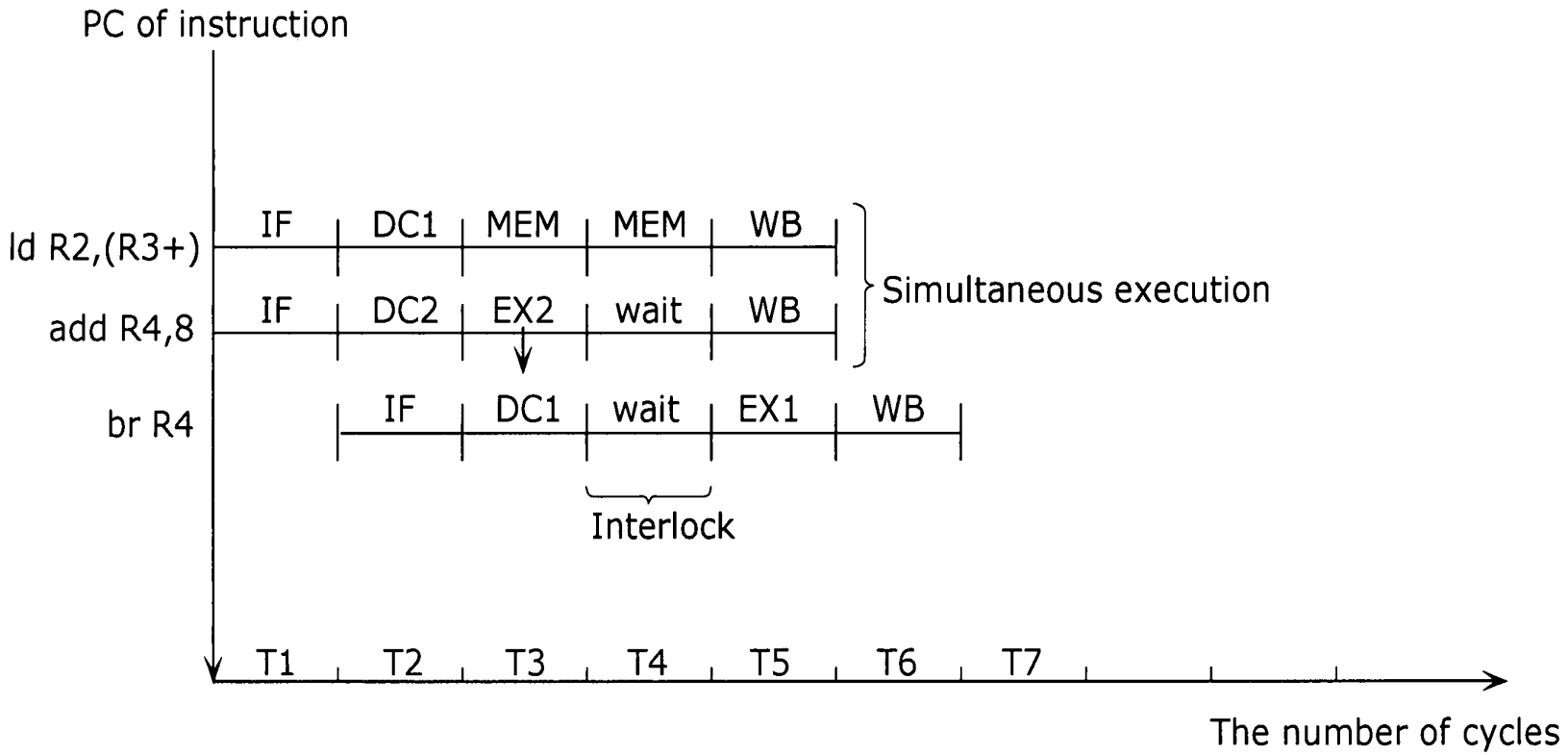
FIG. 8

| | | Address | Mnemonic | Instruction execution contents |
|---|---|---|---|---|
| Simultaneous execution | Instruction 6 | 0x9000 | cmp F0,R0,R1 | F0=1 when R0 equals to R1,<br>F0=0 when R0 does not equal to R1, |
| | Instruction 7 | 0x9002 | [F0] add R2,1 | R2=R2+1 when F0 is 1<br>Nothing is performed when F0 is not 1 |
| | Instruction 8 | 0x9004 | add R3,1 | R3=R3+1 |

FIG. 9

| | | Address | Mnemonic | Instruction execution contents |
|---|---|---|---|---|
| Simultaneous executionl | Instruction 12 | 0xB000 | mov R0,1 | R0=1 |
| | Instruction 13 | 0xB002 | ld R1,(R2+) | R1=mem(R2),R2=R2+4 |
| | Instruction 14 | 0xB004 | mov R1,3 | R1=3 |

FIG. 10


PC of instruction
ld R2,(R3+)
IF
DC1
MEM
MEM
WB
add R4,8
IF
DC2
EX2
wait
WB
Simultaneous execution
br R4
IF
DC1
wait
EX1
WB
Interlock
T1
T2
T3
T4
T5
T6
T7
The number of cycles

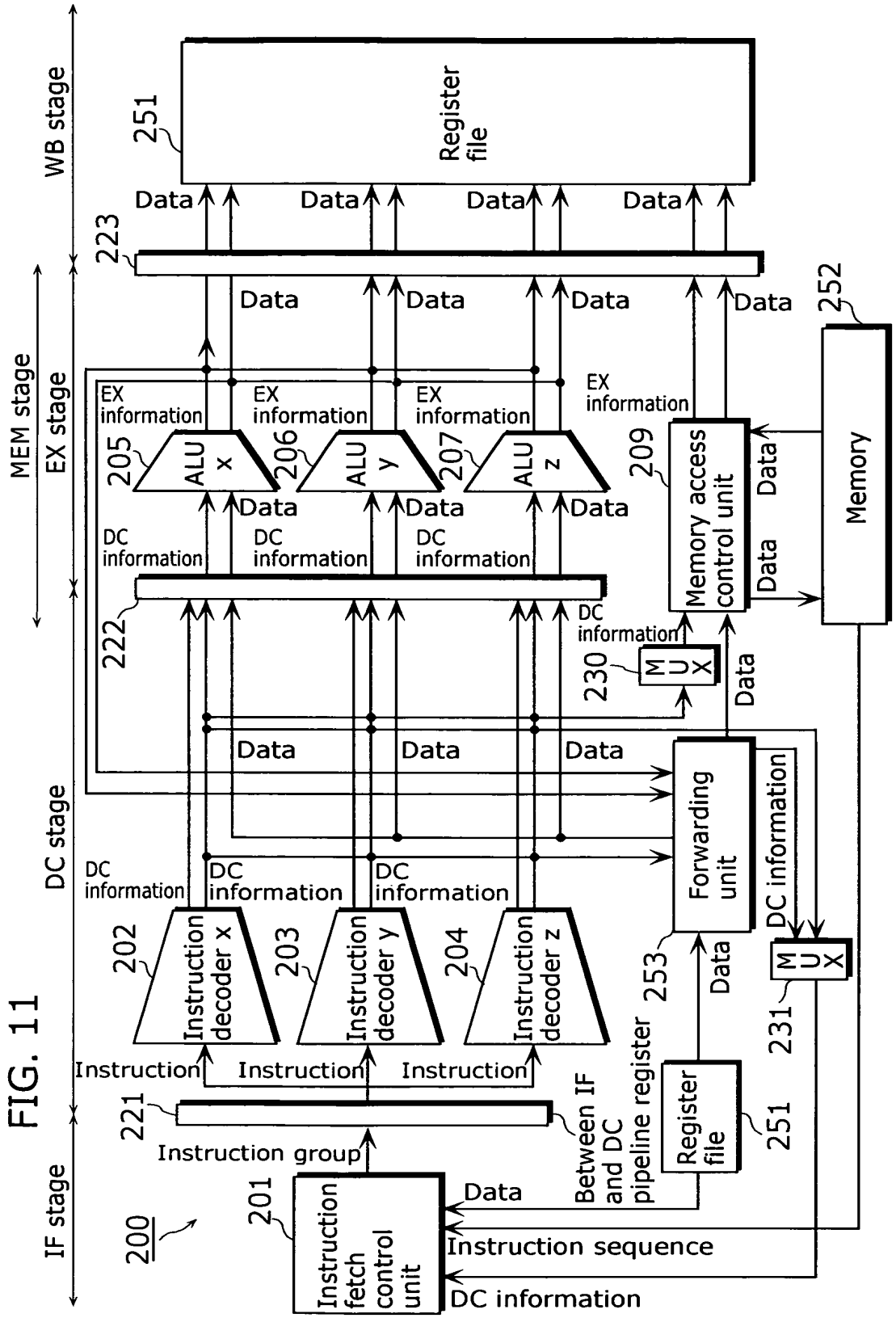

WB stage
MEM stage
EX stage
DC stage
IF stage
Register file
251
Data
223
EX information
ALU x
ALU y
ALU z
205
206
207
209
Memory access control unit
252
Memory
DC information
222
230
MUX
Forwarding unit
202
203
204
Instruction decoder x
Instruction decoder y
Instruction decoder z
Instruction
221
Instruction group
Between IF and DC pipeline register
253
231
200
201
Instruction fetch control unit
Instruction sequence

FIG. 11

FIG. 12

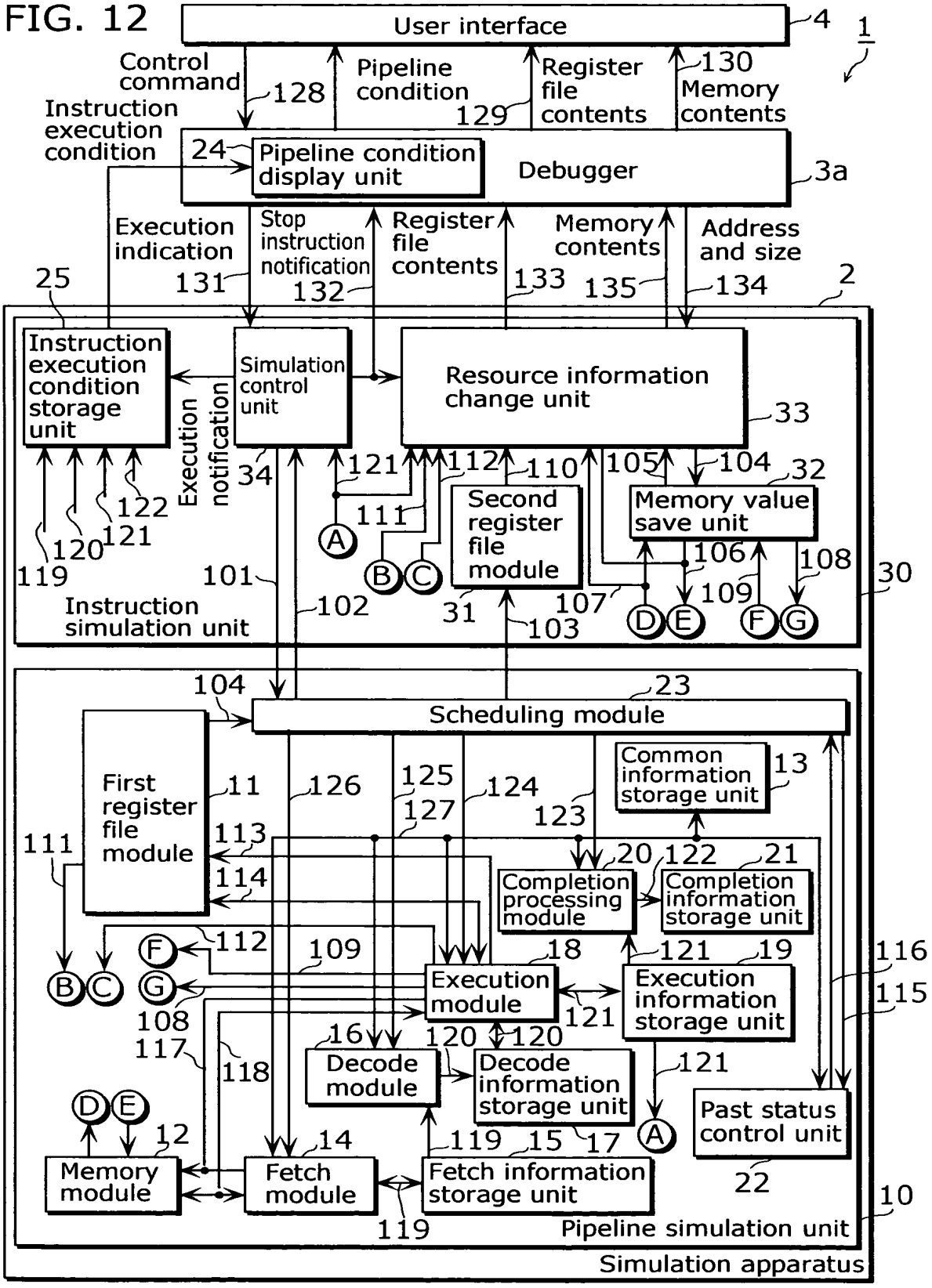

User interface
4
1
Control command
128
Pipeline condition
129
Register file contents
130
Memory contents
Instruction execution condition
24
Pipeline condition display unit
Debugger
3a
Execution indication
131
Stop instruction notification
132
Register file contents
133
Memory contents
135
Address and size
134
25
2
Instruction execution condition storage unit
Simulation control unit
Resource information change unit
33
Execution notification
34
121
112
110
105
104
32
122
120
121
119
111
A
B
C
Second register file module
Memory value save unit
106
108
101
102
31
103
107
109
D
E
F
G
30
Instruction simulation unit
104
23
Scheduling module
First register file module
11
126
125
124
127
123
Common information storage unit
13
111
113
114
20
122
21
Completion processing module
Completion information storage unit
F
112
109
18
121
19
116
B
C
G
108
Execution module
Execution information storage unit
115
117
118
16
120
120
121
Decode module
Decode information storage unit
A
Past status control unit
D
E
12
14
119
15
17
Memory module
Fetch module
Fetch information storage unit
22
119
10
Pipeline simulation unit
Simulation apparatus

FIG. 13

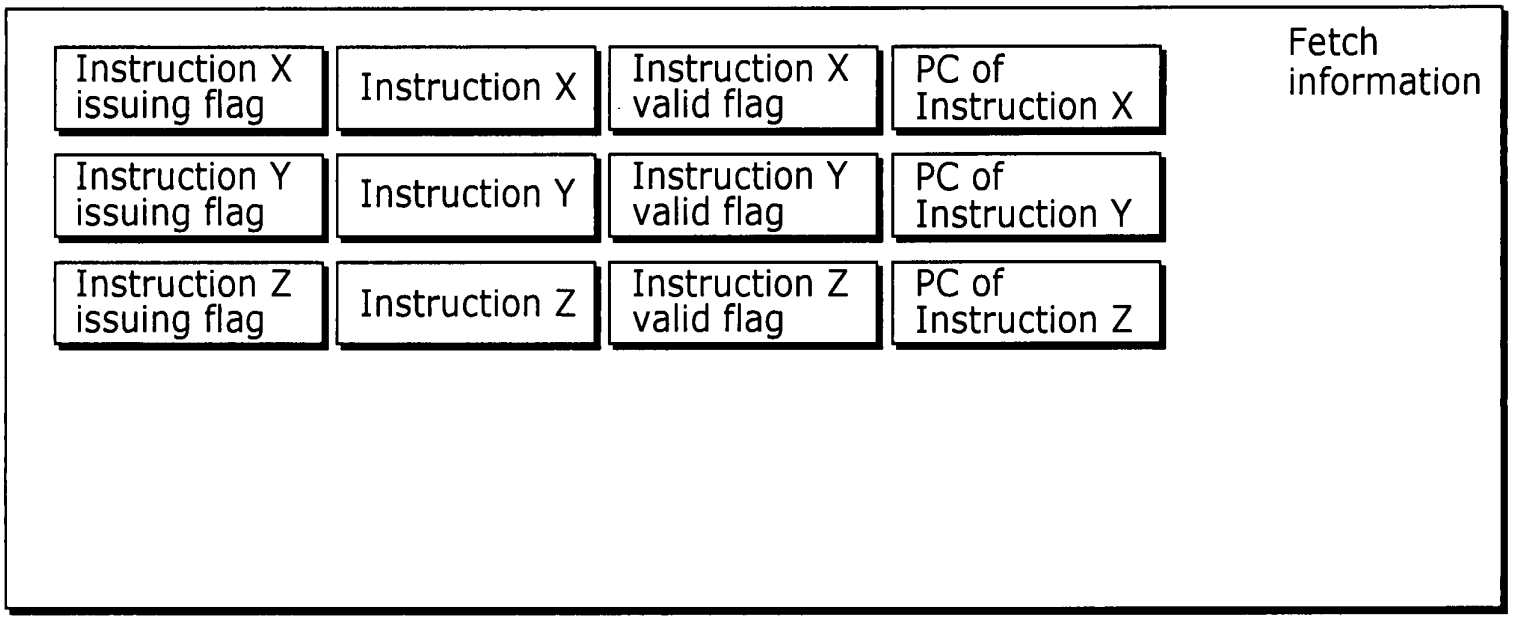

Instruction X issuing flag
Instruction X
Instruction X valid flag
PC of Instruction X
Instruction Y issuing flag
Instruction Y
Instruction Y valid flag
PC of Instruction Y
Instruction Z issuing flag
Instruction Z
Instruction Z valid flag
PC of Instruction Z
Fetch information

FIG. 14

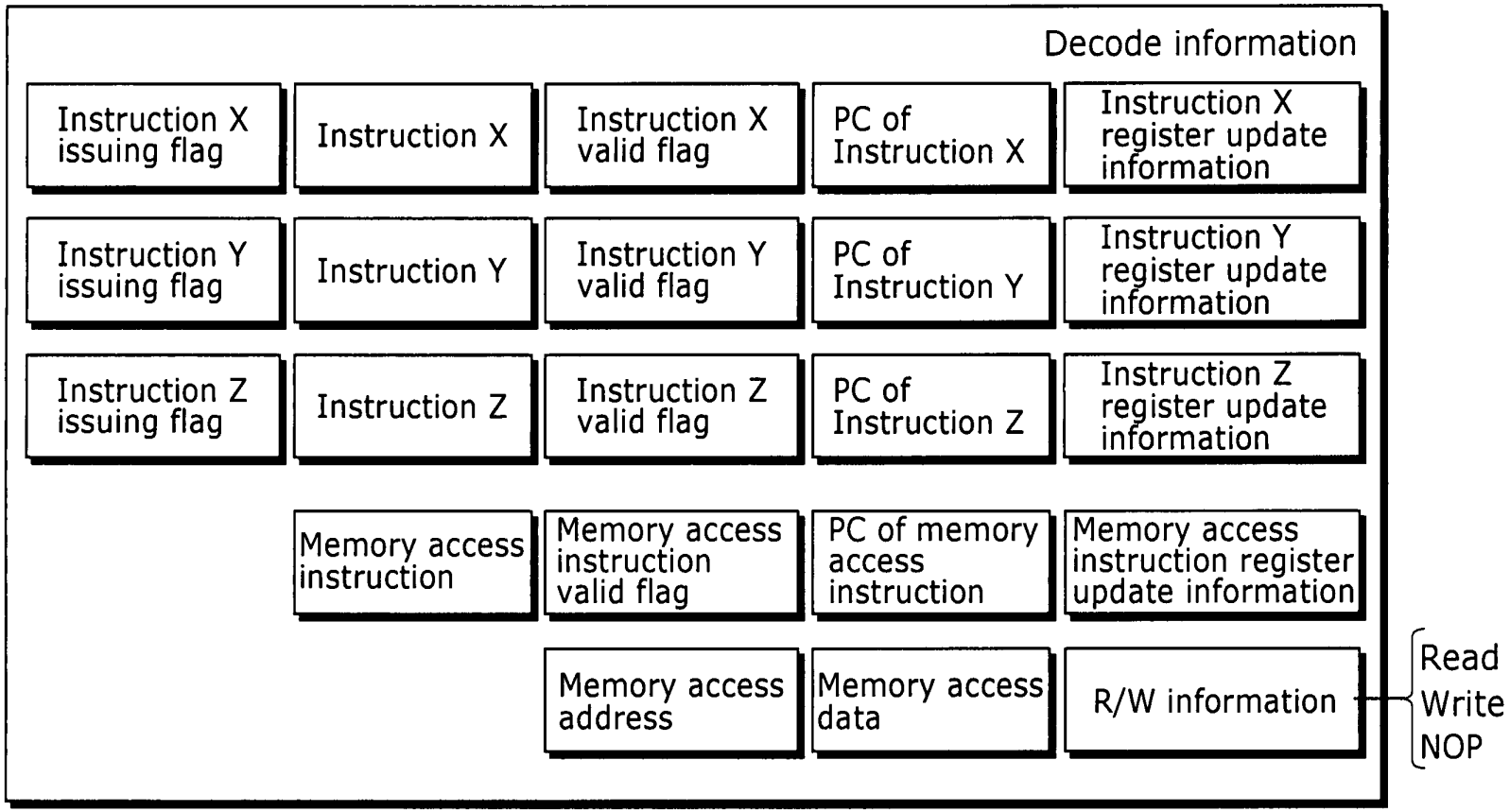

Decode information
Instruction X issuing flag
Instruction X
Instruction X valid flag
PC of Instruction X
Instruction X register update information
Instruction Y issuing flag
Instruction Y
Instruction Y valid flag
PC of Instruction Y
Instruction Y register update information
Instruction Z issuing flag
Instruction Z
Instruction Z valid flag
PC of Instruction Z
Instruction Z register update information
Memory access instruction
Memory access instruction valid flag
PC of memory access instruction
Memory access instruction register update information
Memory access address
Memory access data
R/W information
Read
Write
NOP

FIG. 15

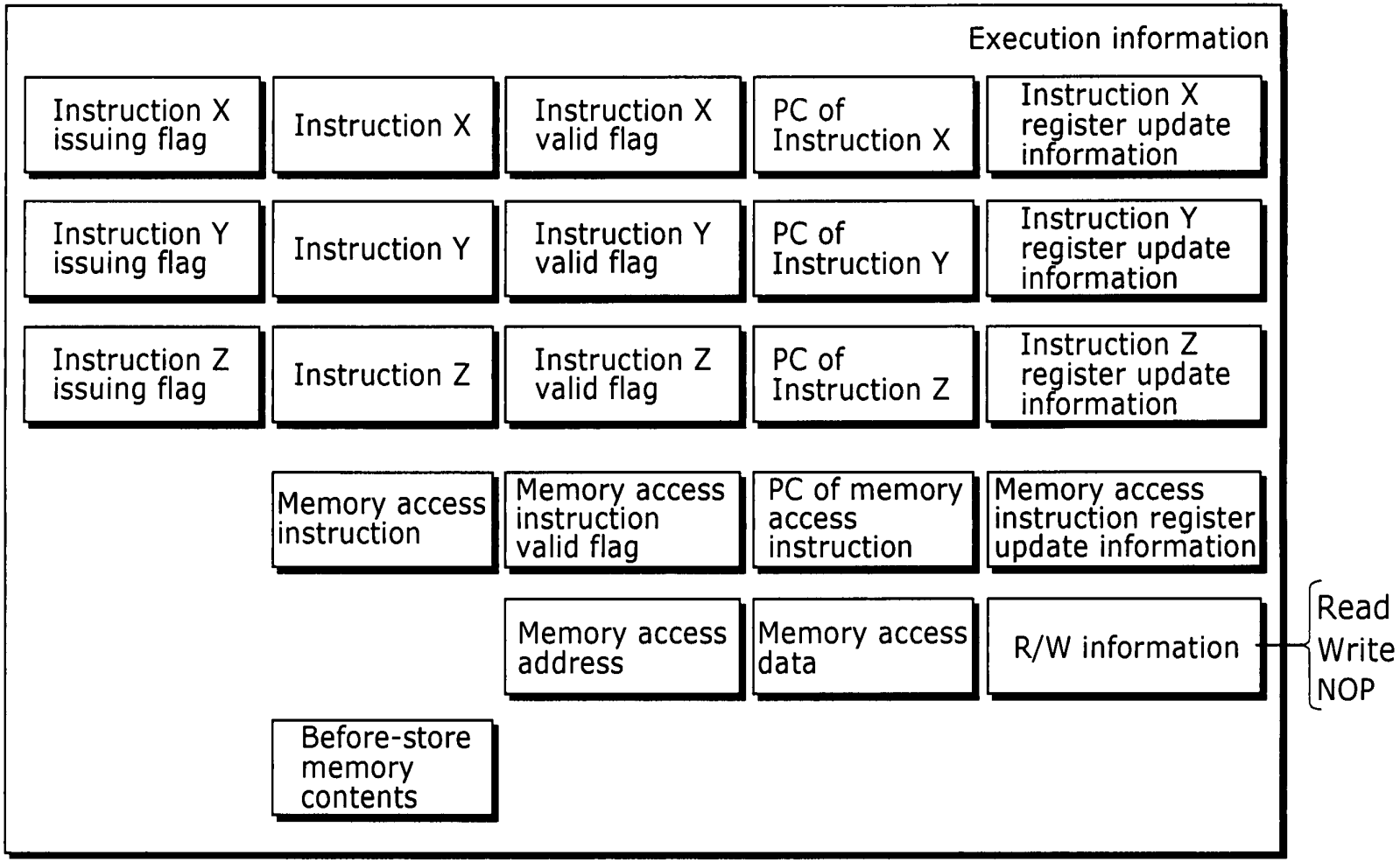

Execution information
Instruction X issuing flag
Instruction X
Instruction X valid flag
PC of Instruction X
Instruction X register update information
Instruction Y issuing flag
Instruction Y
Instruction Y valid flag
PC of Instruction Y
Instruction Y register update information
Instruction Z issuing flag
Instruction Z
Instruction Z valid flag
PC of Instruction Z
Instruction Z register update information
Memory access instruction
Memory access instruction valid flag
PC of memory access instruction
Memory access instruction register update information
Memory access address
Memory access data
R/W information
Read
Write
NOP
Before-store memory contents

FIG. 16

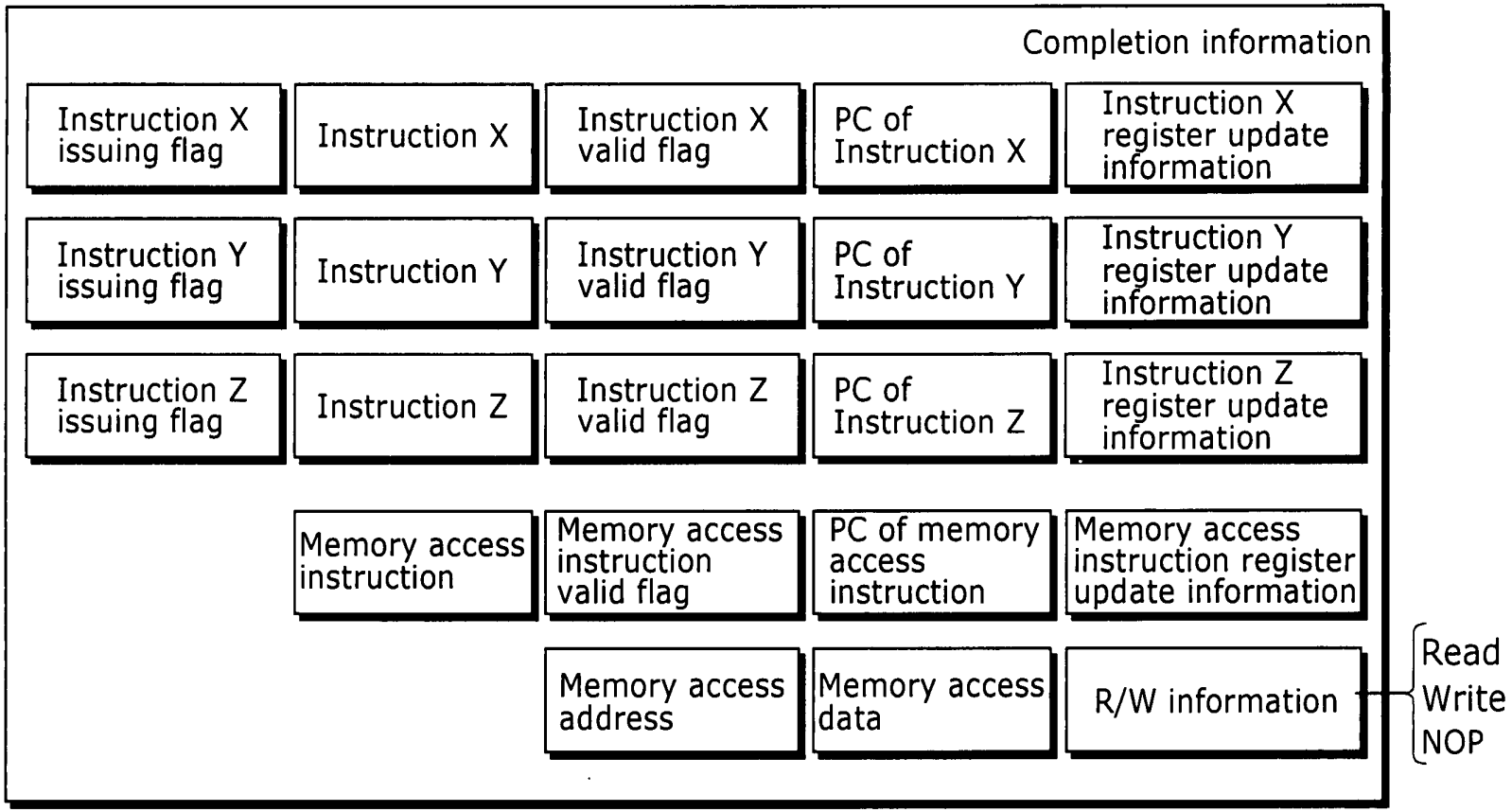

Completion information
Instruction X issuing flag
Instruction X
Instruction X valid flag
PC of Instruction X
Instruction X register update information
Instruction Y issuing flag
Instruction Y
Instruction Y valid flag
PC of Instruction Y
Instruction Y register update information
Instruction Z issuing flag
Instruction Z
Instruction Z valid flag
PC of Instruction Z
Instruction Z register update information
Memory access instruction
Memory access instruction valid flag
PC of memory access instruction
Memory access instruction register update information
Memory access address
Memory access data
R/W information
Read
Write
NOP

FIG. 17

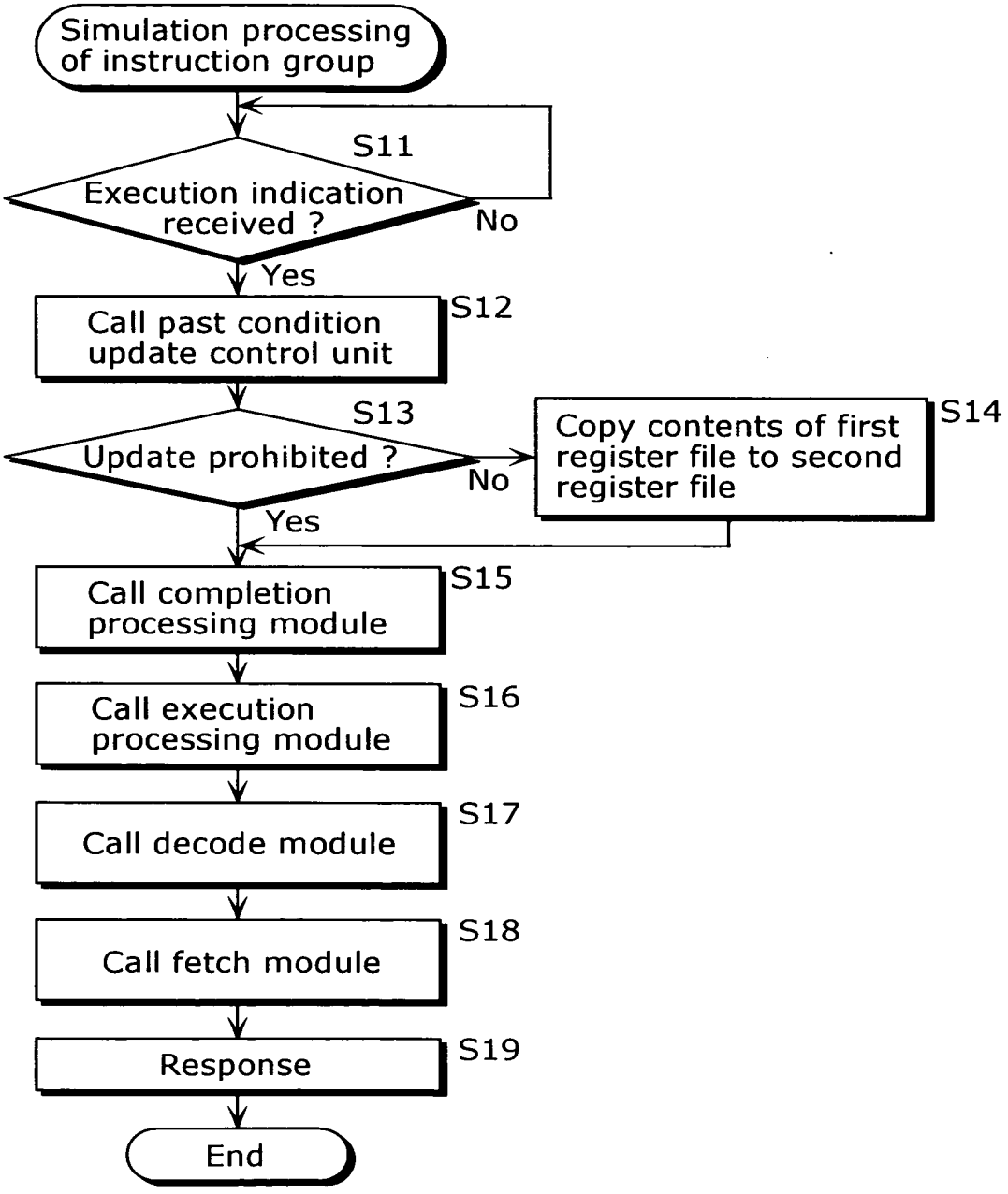

Simulation processing of instruction group
S11
Execution indication received ?
No
Yes
Call past condition update control unit
S12
S13
Update prohibited ?
No
Copy contents of first register file to second register file
S14
Yes
Call completion processing module
S15
Call execution processing module
S16
Call decode module
S17
Call fetch module
S18
Response
S19
End

FIG. 18

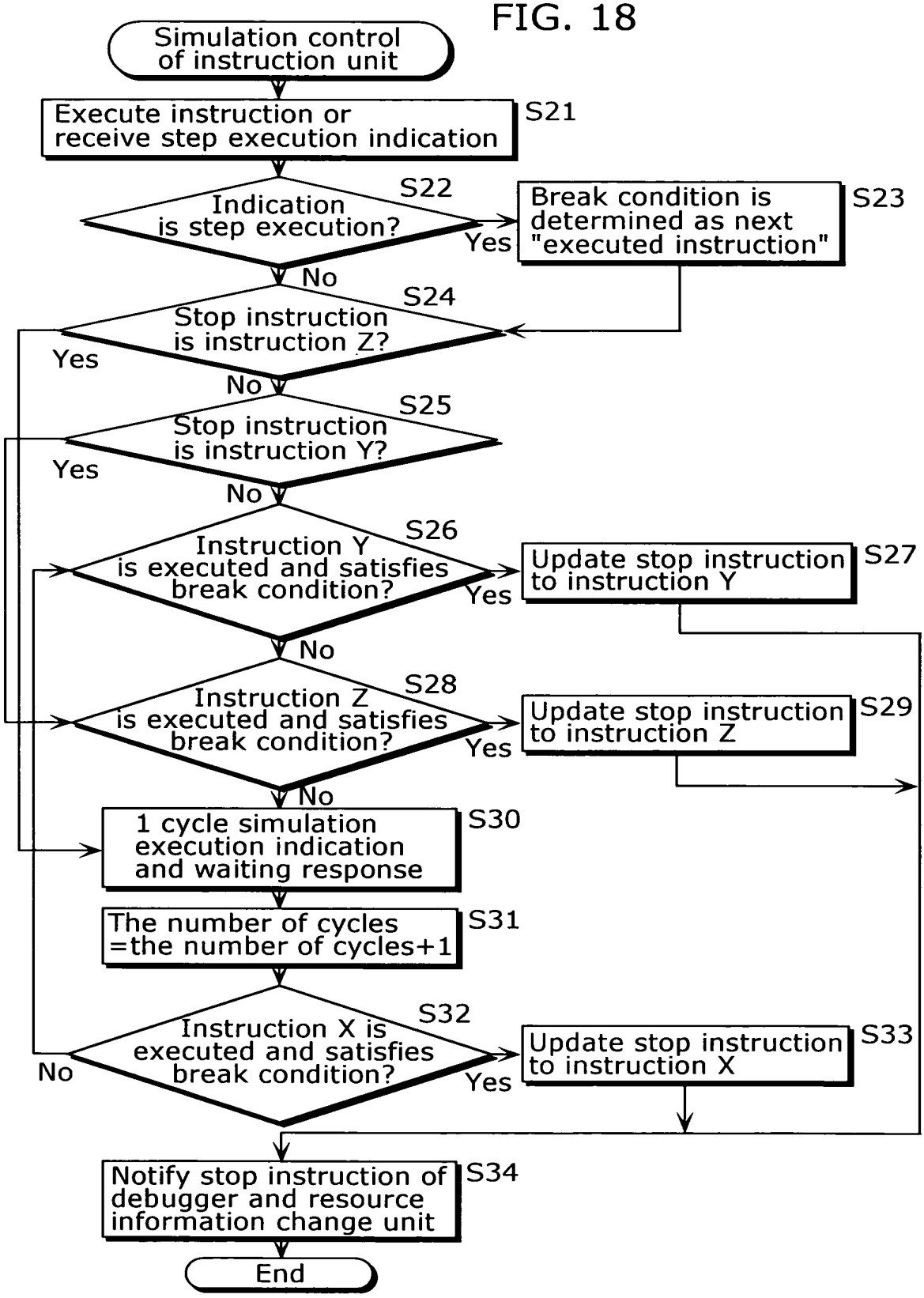

Simulation control of instruction unit
Execute instruction or receive step execution indication
S21
Indication is step execution?
S22
Yes
Break condition is determined as next "executed instruction"
S23
No
Stop instruction is instruction Z?
S24
Yes
No
Stop instruction is instruction Y?
S25
Yes
No
Instruction Y is executed and satisfies break condition?
S26
Yes
Update stop instruction to instruction Y
S27
No
Instruction Z is executed and satisfies break condition?
S28
Yes
Update stop instruction to instruction Z
S29
No
1 cycle simulation execution indication and waiting response
S30
The number of cycles =the number of cycles+1
S31
Instruction X is executed and satisfies break condition?
S32
No
Yes
Update stop instruction to instruction X
S33
Notify stop instruction of debugger and resource information change unit
S34
End

FIG. 19

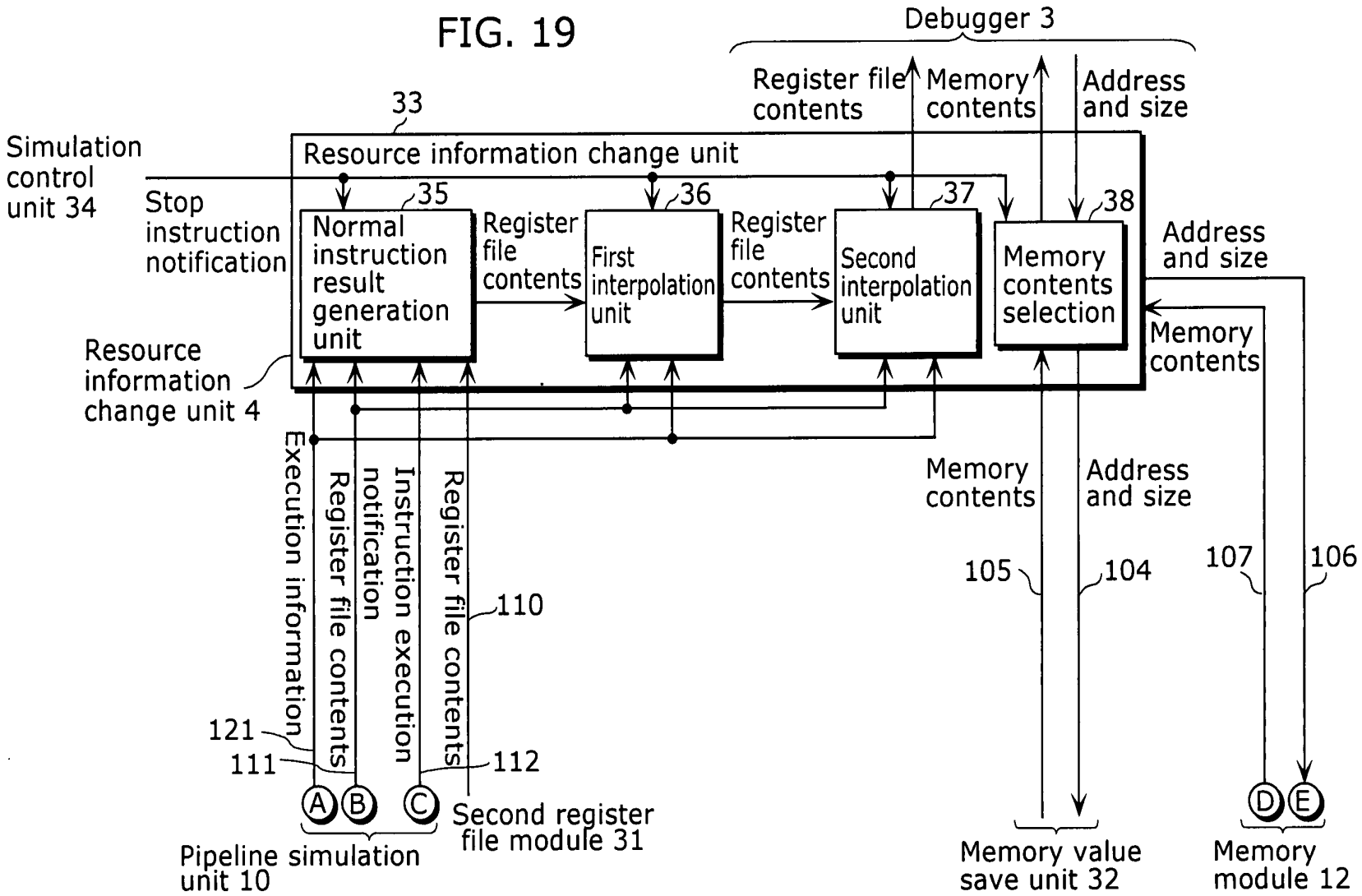

Debugger 3
Register file contents
Memory contents
Address and size
33
Resource information change unit
Simulation control unit 34
Stop instruction notification
35
Normal instruction result generation unit
Register file contents
36
First interpolation unit
Register file contents
37
Second interpolation unit
38
Memory contents selection
Address and size
Memory contents
Resource information change unit 4
Execution information
Register file contents
Instruction execution notification
Register file contents
110
121
111
112
A
B
C
Second register file module 31
Pipeline simulation unit 10
Memory contents
Address and size
105
104
107
106
D
E
Memory value save unit 32
Memory module 12

FIG. 20

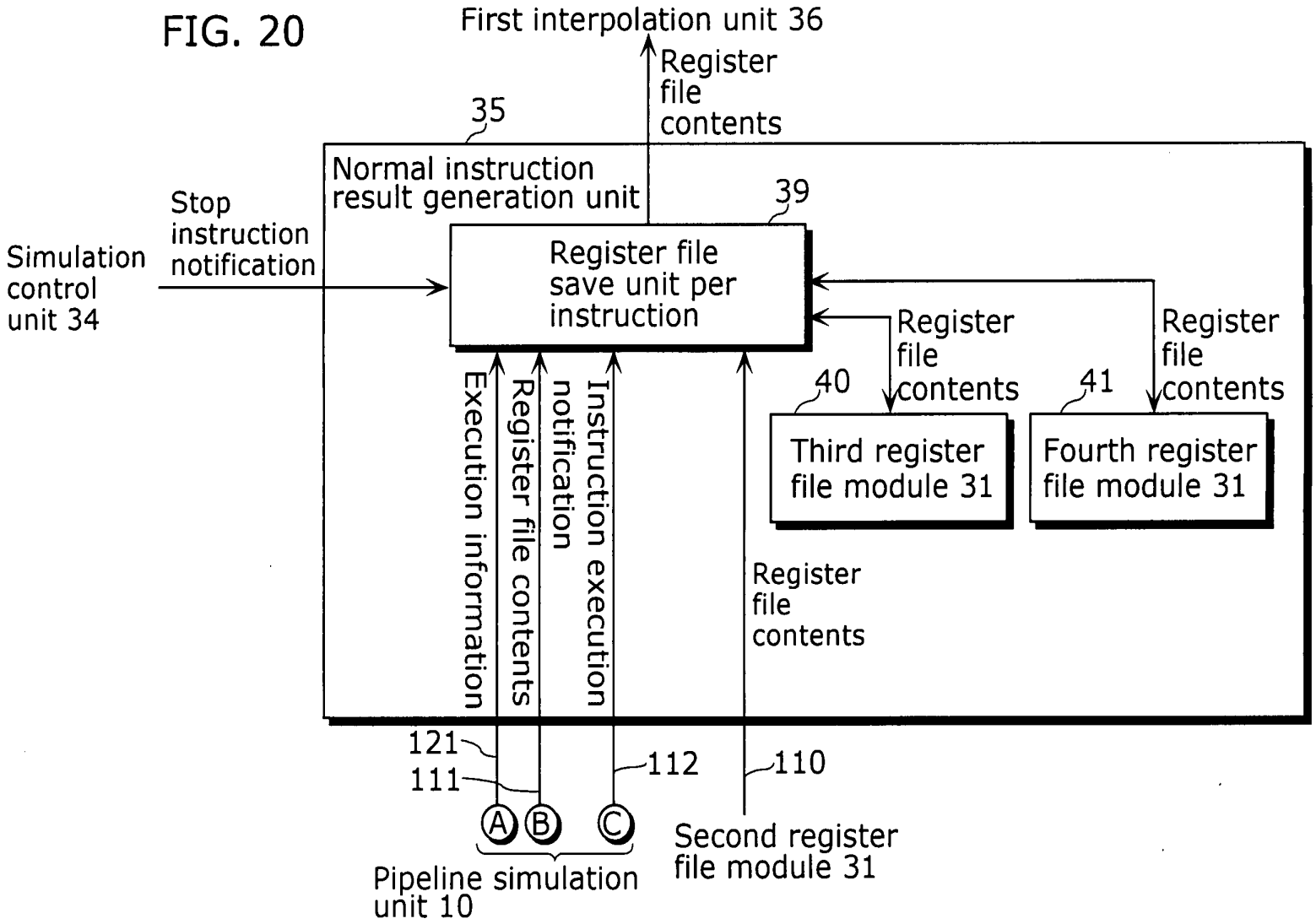

First interpolation unit 36
Register file contents
35
Normal instruction result generation unit
39
Stop instruction notification
Simulation control unit 34
Register file save unit per instruction
Register file contents
Register file contents
40
41
Third register file module 31
Fourth register file module 31
Execution information
Register file contents
Instruction execution notification
Register file contents
121
111
112
110
A
B
C
Second register file module 31
Pipeline simulation unit 10

FIG. 21

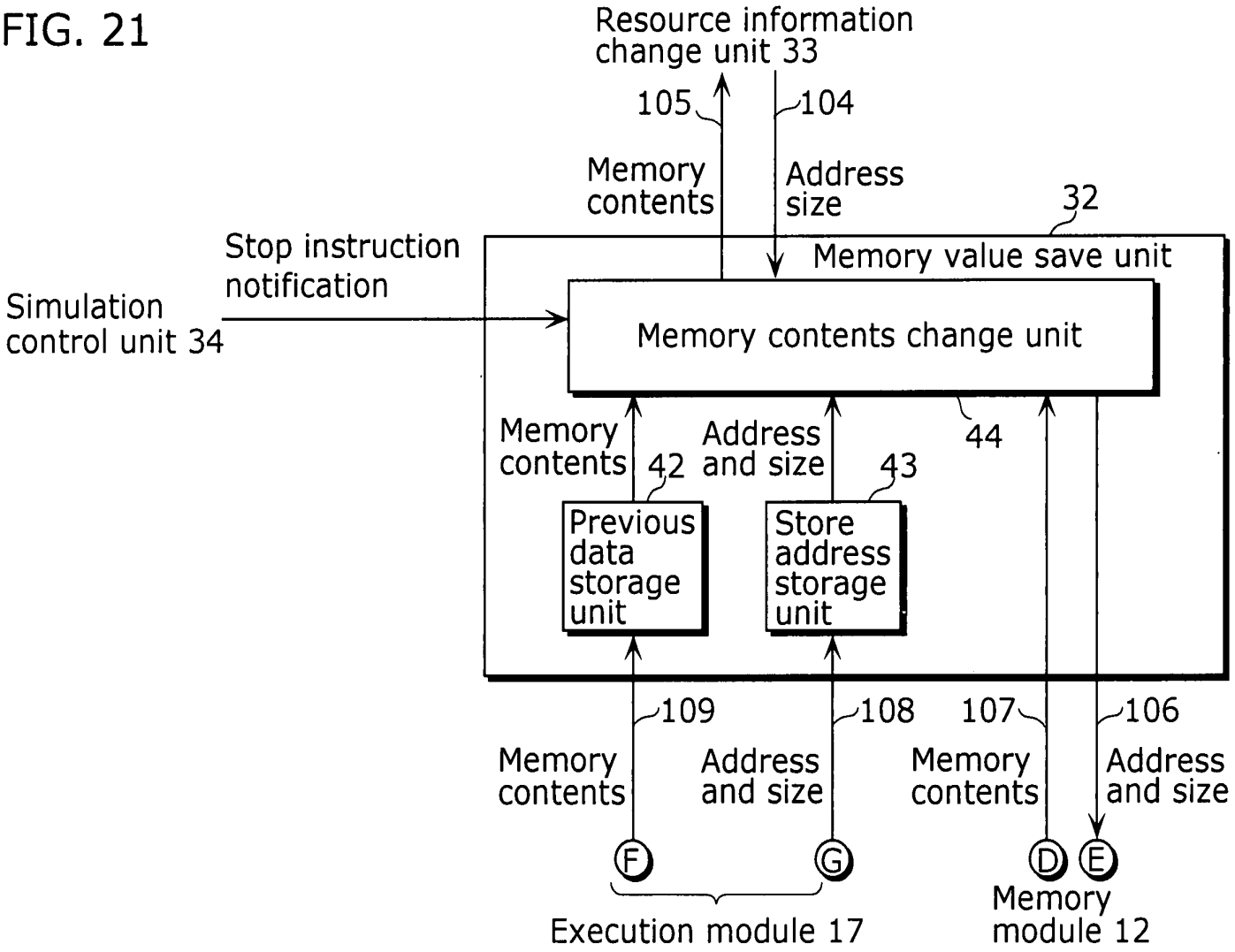

Resource information change unit 33
105
104
Memory contents
Address size
32
Stop instruction notification
Simulation control unit 34
Memory value save unit
Memory contents change unit
Memory contents
42
Address and size
43
44
Previous data storage unit
Store address storage unit
109
108
107
106
Memory contents
Address and size
Memory contents
Address and size
F
G
D
E
Execution module 17
Memory module 12

FIG. 22

| | | PC | Mnemonic | Simulation result {R0,R1,R2,R3,F0} | Display result {R0,R1,R2,R3,F0} | Stop |
|---|---|---|---|---|---|---|
| Instruction group 1 | Instruction 6 | 0x9000 | cmp F0,R0,R1 | [1,0,0,0,0] | [1,0,0,0,1] | ○ |
| | Instruction 7 | 0x9002 | [F0] add R2,1 | [1,0,0,0,0] | [1,0,0,0,0] | × |
| Instruction group 2 | Instruction 8 | 0x9004 | add R3,1 | [1,0,0,1,0] | [1,0,0,0,0] | ○ |

FIG. 23

| | | PC | Mnemonic | Display result {R0,R1,R2,R3,R4,R5,R6} | Stop |
|---|---|---|---|---|---|
| Instruction group 1 | Instruction 1 | 0x8000 | sub R0,R1 | [10,5,0,0,0,1,2] | ○ |
| | Instruction 2 | 0x8002 | add R2,1 | [5,5,0,0,0,1,2] | ○ |
| | Instruction 3 | 0x8004 | ld R3,(R4+) | [5,5,1,0,0,1,2] | ○ |
| Instruction group 2 | Instruction 4 | 0x8006 | st (R4+),R2 | [5,5,1,100,4,1,2] | ○ |
| | Instruction 5 | 0x8008 | or R5,R6 | [5,5,1,100,8,1,2] | ○ |

FIG. 24

| | | PC | Mnemonic | Display result {R0,R1,R2} | Stop |
|---|---|---|---|---|---|
| Instruction group 5 | Instruction 12 | 0xB000 | mov R0,1 | [0,0,0] | ○ |
| | Instruction 13 | 0xB002 | ld R1,(R2+) | [1,0,0] | ○ |
| | Instruction 14 | 0xB004 | mov R1,3 | [1,200,4] | ○ |

| First register file [R0,R1,R2] | Second register file [R0,R1,R2] | Third register file [R0,R1,R2] | Fourth register file [R0,R1,R2] | Memory access data |
|---|---|---|---|---|
| {1,3,4} | {0,0,0} | {1,0,0} | {1,0,4} | {200} |

FIG. 25

| | | PC | Mnemonic | Simulation result {R0,R1,R2,R3,F0} | Display result {R0,R1,R2,R3,F0} | Stop |
|---|---|---|---|---|---|---|
| Instruction group 1 | Instruction 6 | 0x9000 | cmp F0, R0, R1 | [1,0,0,0,0] | [1,0,0,0,0] | ○ |
| | Instruction 7 | 0x9002 | [F0] add R2, 1 | [1,0,0,0,0] | [1,0,0,0,0] | ○ |
| Instruction group 2 | Instruction 8 | 0x9004 | add R3, 1 | [1,0,0,1,0] | [1,0,0,1,0] | ○ |

FIG. 26

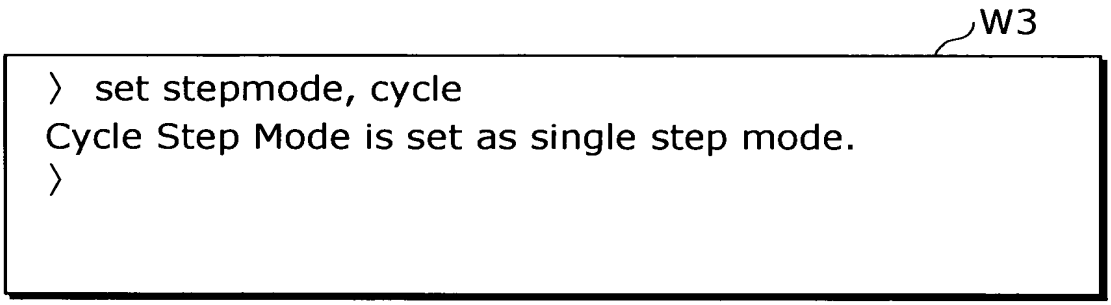

W3
〉 set stepmode, cycle
Cycle Step Mode is set as single step mode.
〉

FIG. 27

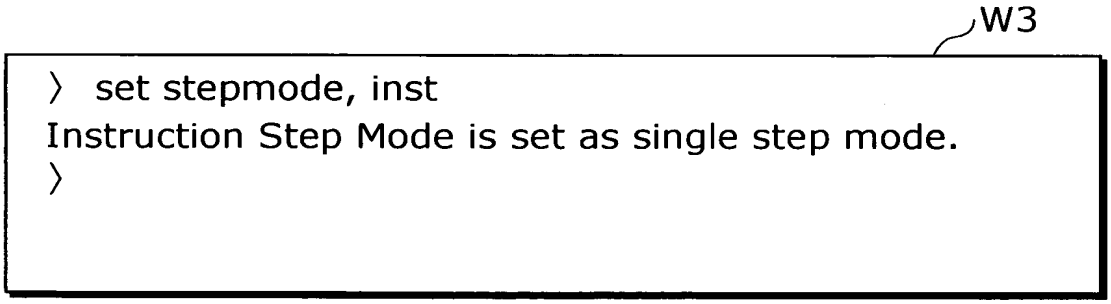

W3
〉 set stepmode, inst
Instruction Step Mode is set as single step mode.
〉

FIG. 28

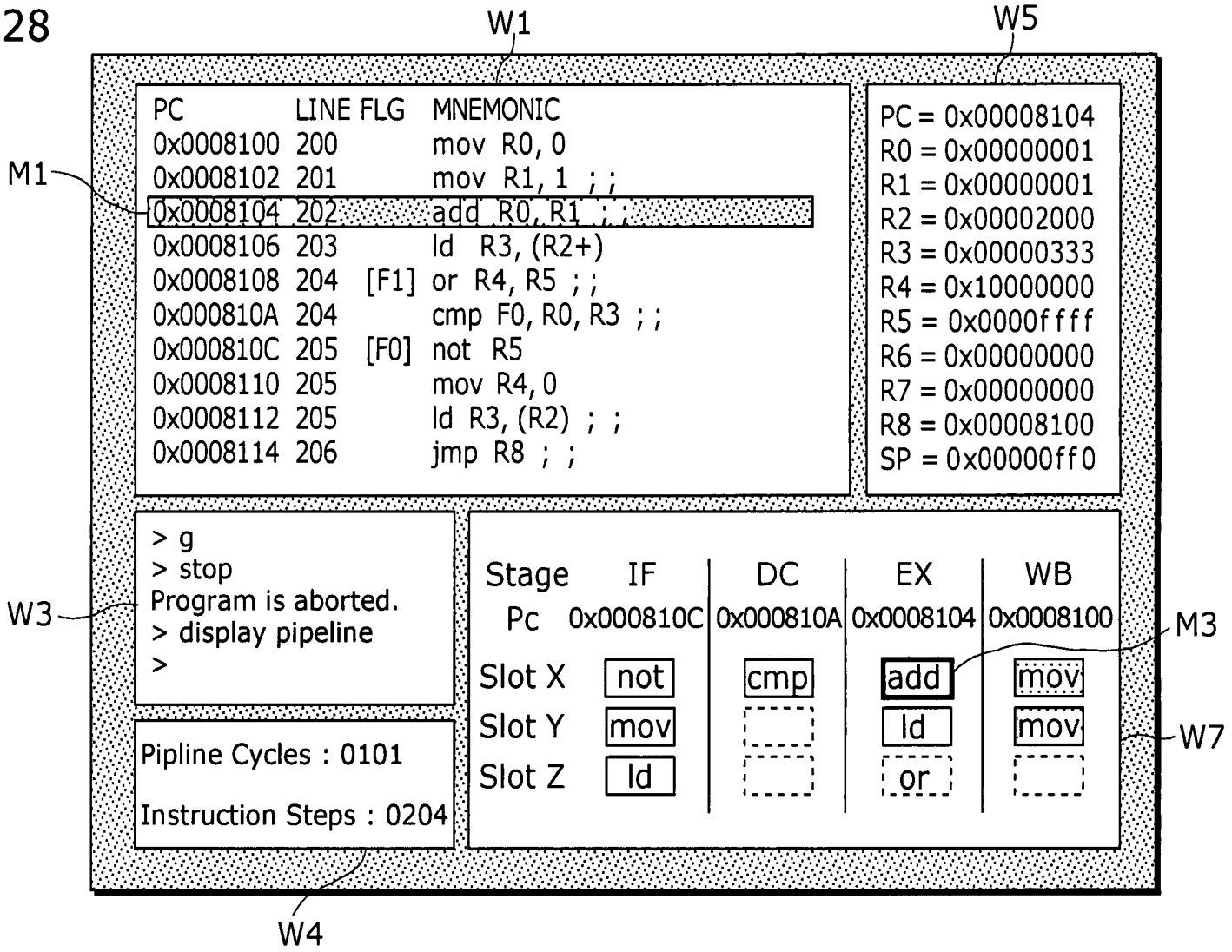

W1
W5
M1
PC LINE FLG MNEMONIC
0x0008100 200 mov R0, 0
0x0008102 201 mov R1, 1 ; ;
0x0008104 202 add R0, R1 ; ;
0x0008106 203 ld R3, (R2+)
0x0008108 204 [F1] or R4, R5 ; ;
0x000810A 204 cmp F0, R0, R3 ; ;
0x000810C 205 [F0] not R5
0x0008110 205 mov R4, 0
0x0008112 205 ld R3, (R2) ; ;
0x0008114 206 jmp R8 ; ;
PC = 0x00008104
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002000
R3 = 0x00000333
R4 = 0x10000000
R5 = 0x0000ffff
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00008100
SP = 0x00000ff0
W3
> g
> stop
Program is aborted.
> display pipeline
>
Stage IF DC EX WB
Pc 0x000810C 0x000810A 0x0008104 0x0008100
M3
Slot X not cmp add mov
Slot Y mov ld mov
Slot Z ld or
W7
Pipline Cycles : 0101
Instruction Steps : 0204
W4

FIG. 29

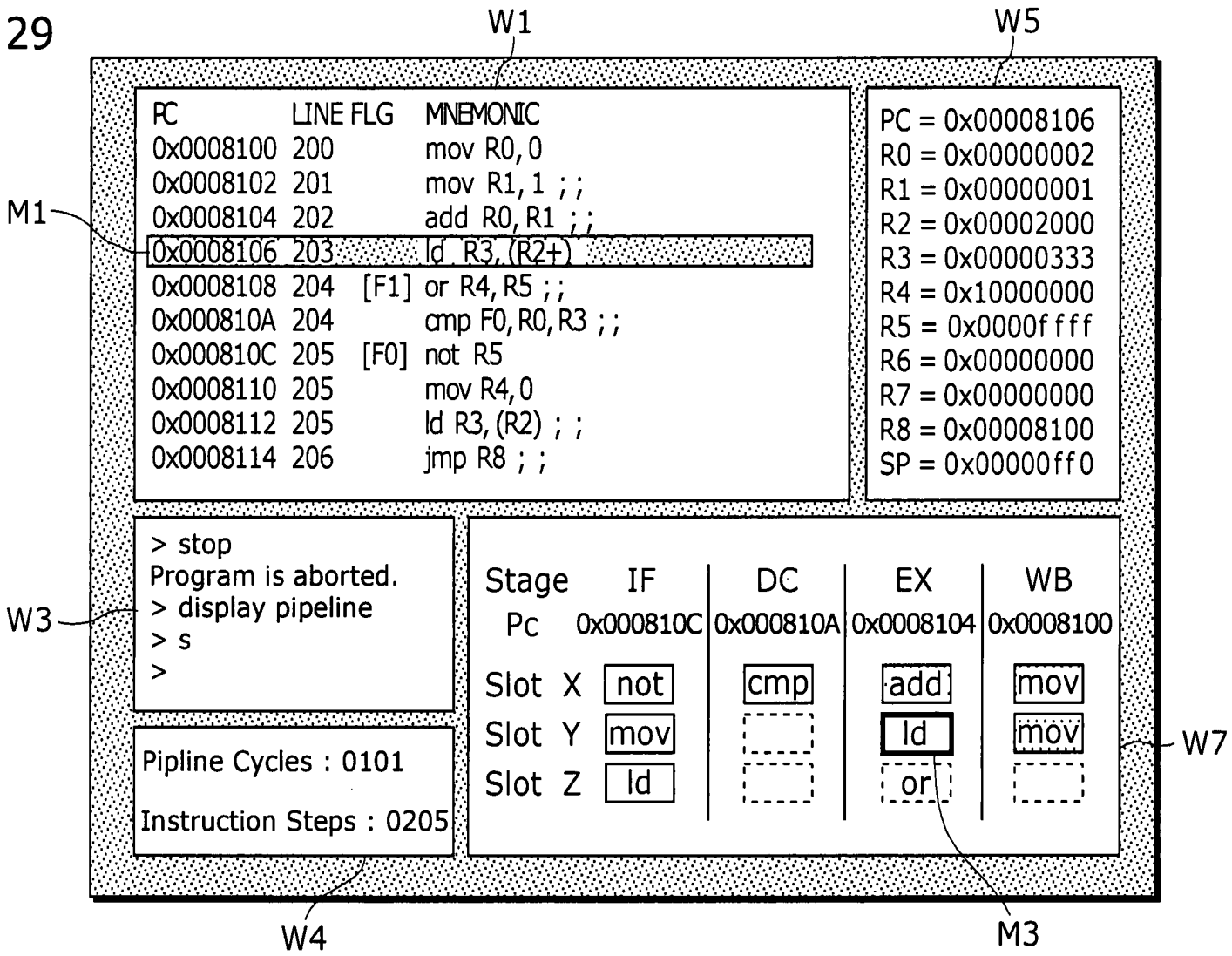

W1
W5
M1
PC LINE FLG MNEMONIC
0x0008100 200 mov R0, 0
0x0008102 201 mov R1, 1 ; ;
0x0008104 202 add R0, R1 ; ;
0x0008106 203 ld R3, (R2+)
0x0008108 204 [F1] or R4, R5 ; ;
0x000810A 204 cmp F0, R0, R3 ; ;
0x000810C 205 [F0] not R5
0x0008110 205 mov R4, 0
0x0008112 205 ld R3, (R2) ; ;
0x0008114 206 jmp R8 ; ;
PC = 0x00008106
R0 = 0x00000002
R1 = 0x00000001
R2 = 0x00002000
R3 = 0x00000333
R4 = 0x10000000
R5 = 0x0000ffff
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00008100
SP = 0x00000ff0
W3
> stop
Program is aborted.
> display pipeline
> s
>
Stage IF DC EX WB
Pc 0x000810C 0x000810A 0x0008104 0x0008100
Slot X not cmp add mov
Slot Y mov ld mov
Slot Z ld or
W7
Pipline Cycles : 0101
Instruction Steps : 0205
W4
M3

FIG. 30

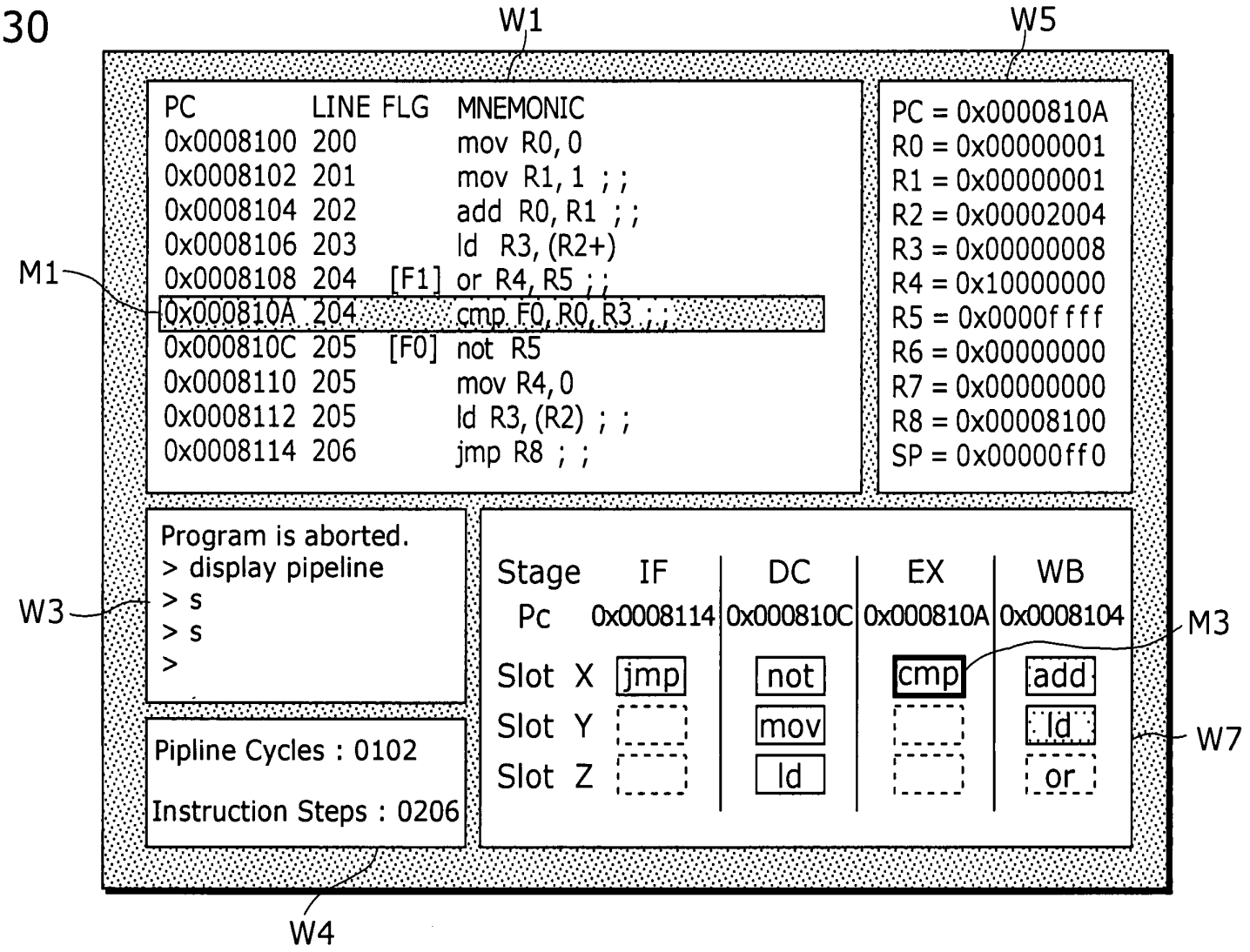

W1
W5
M1
PC LINE FLG MNEMONIC
0x0008100 200 mov R0, 0
0x0008102 201 mov R1, 1 ; ;
0x0008104 202 add R0, R1 ; ;
0x0008106 203 ld R3, (R2+)
0x0008108 204 [F1] or R4, R5 ; ;
0x000810A 204 cmp F0, R0, R3 ; ;
0x000810C 205 [F0] not R5
0x0008110 205 mov R4, 0
0x0008112 205 ld R3, (R2) ; ;
0x0008114 206 jmp R8 ; ;
PC = 0x0000810A
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002004
R3 = 0x00000008
R4 = 0x10000000
R5 = 0x0000ffff
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00008100
SP = 0x00000ff0
W3
Program is aborted.
> display pipeline
> s
> s
>
Stage IF DC EX WB
Pc 0x0008114 0x000810C 0x000810A 0x0008104
M3
Slot X jmp not cmp add
Slot Y mov ld
Slot Z ld or
W7
Pipline Cycles : 0102
Instruction Steps : 0206
W4

FIG. 31

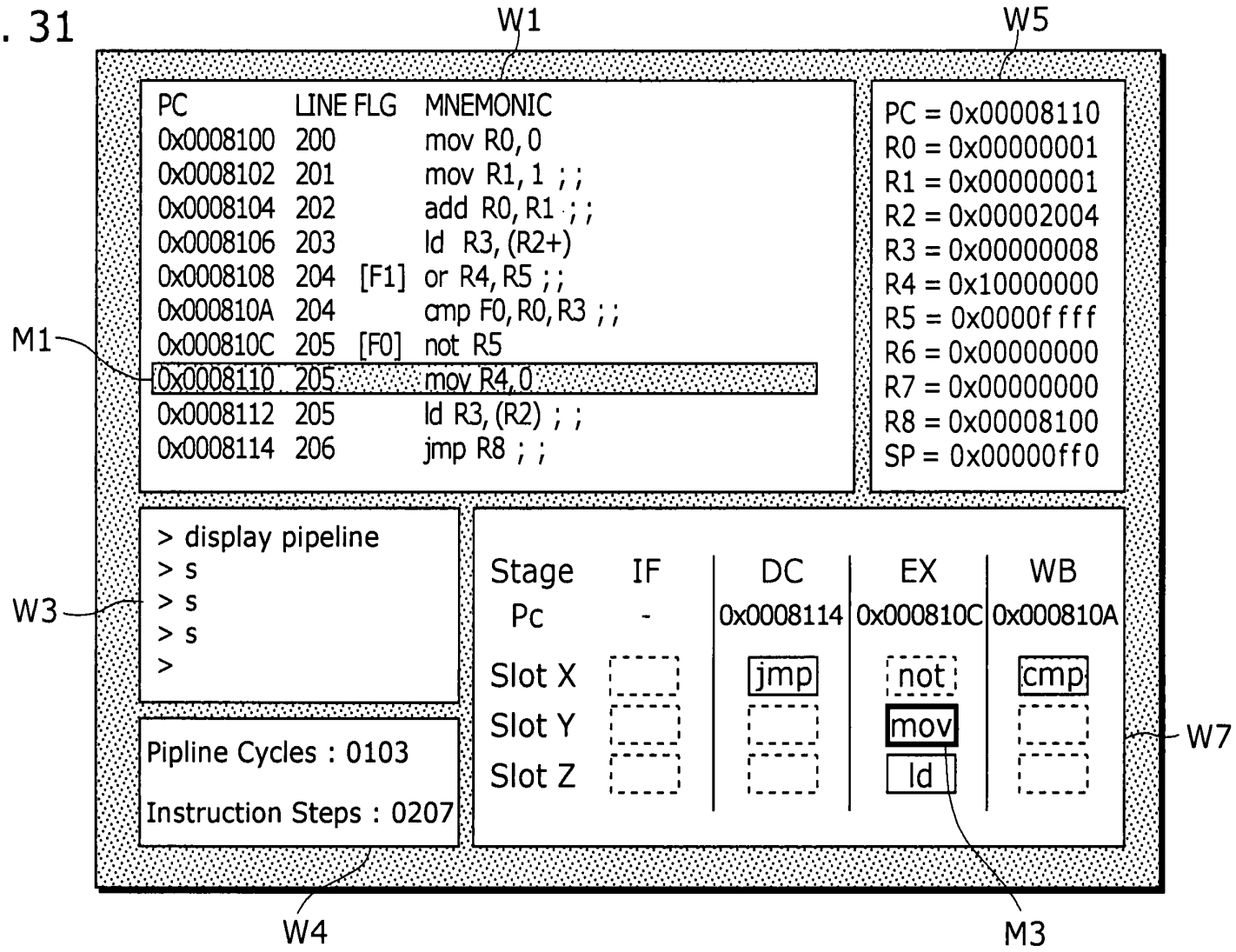

W1
W5
PC LINE FLG MNEMONIC
0x0008100 200 mov R0, 0
0x0008102 201 mov R1, 1 ; ;
0x0008104 202 add R0, R1 ; ;
0x0008106 203 ld R3, (R2+)
0x0008108 204 [F1] or R4, R5 ; ;
0x000810A 204 cmp F0, R0, R3 ; ;
0x000810C 205 [F0] not R5
0x0008110 205 mov R4, 0
0x0008112 205 ld R3, (R2) ; ;
0x0008114 206 jmp R8 ; ;
PC = 0x00008110
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002004
R3 = 0x00000008
R4 = 0x10000000
R5 = 0x0000ffff
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00008100
SP = 0x00000ff0
M1
> display pipeline
> s
> s
> s
>
W3
Stage IF DC EX WB
Pc - 0x0008114 0x000810C 0x000810A
Slot X jmp not cmp
Slot Y mov
Slot Z ld
W7
Pipline Cycles : 0103
Instruction Steps : 0207
W4
M3

FIG. 32

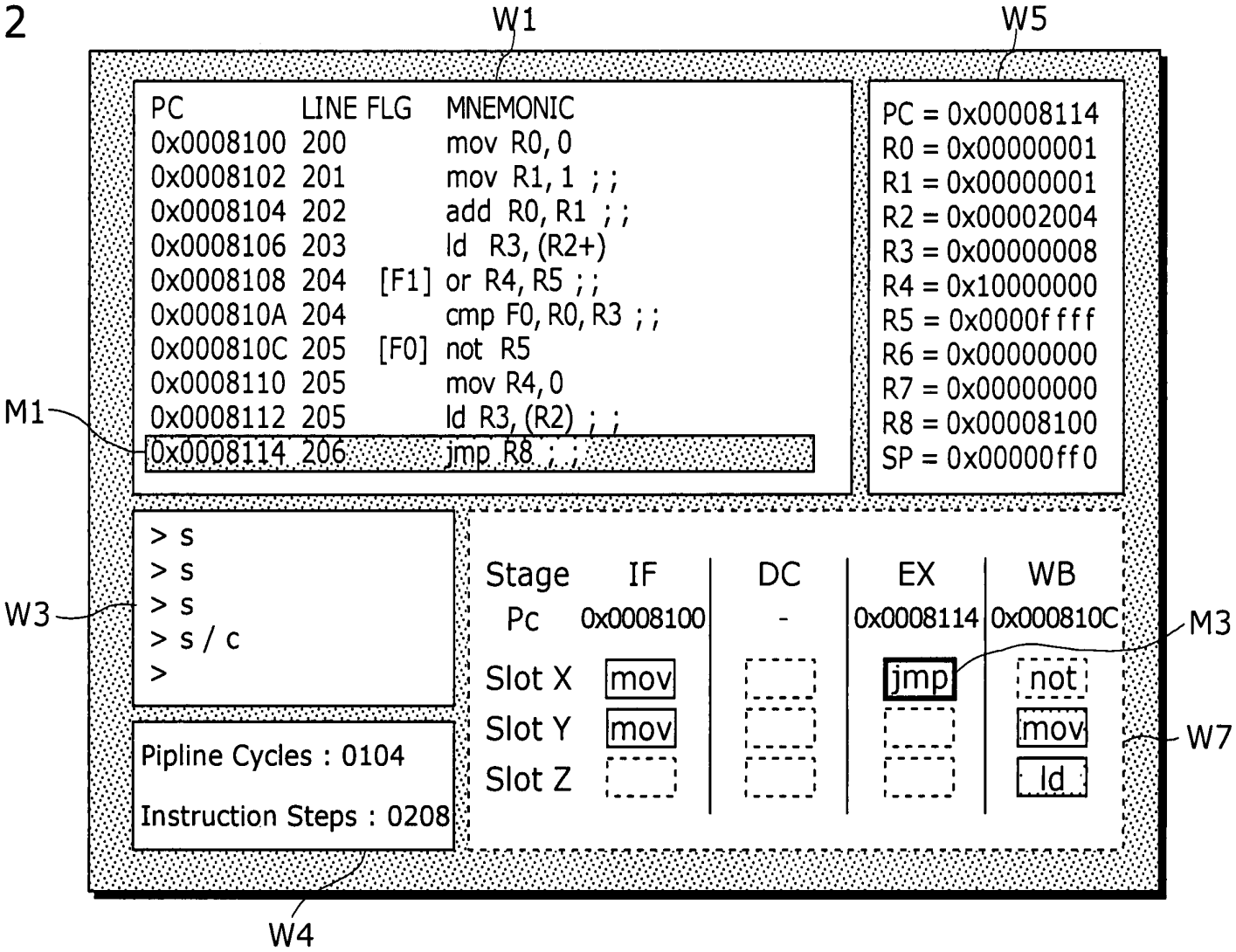
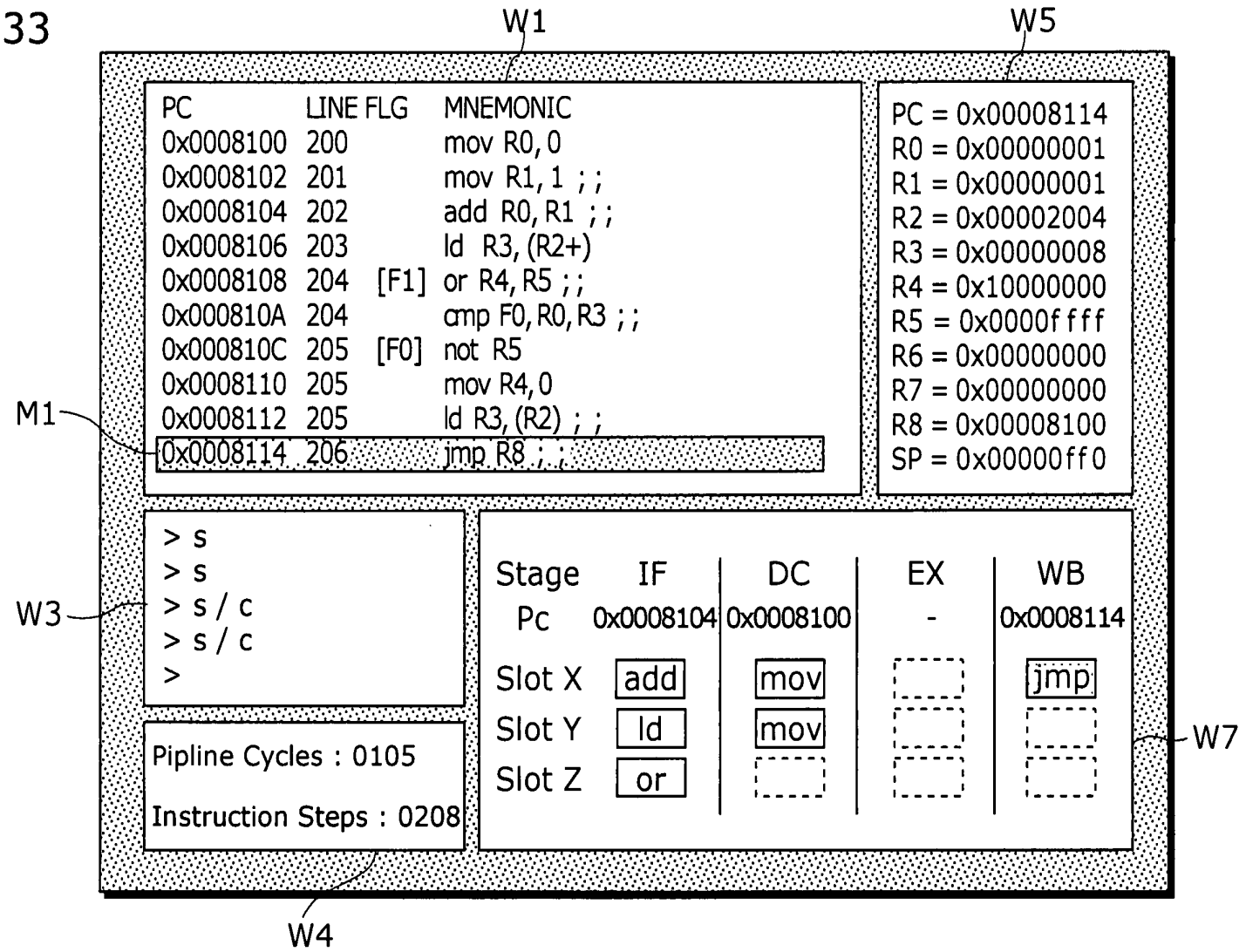

W1
W5
PC LINE FLG MNEMONIC
0x0008100 200 mov R0, 0
0x0008102 201 mov R1, 1 ; ;
0x0008104 202 add R0, R1 ; ;
0x0008106 203 ld R3, (R2+)
0x0008108 204 [F1] or R4, R5 ; ;
0x000810A 204 cmp F0, R0, R3 ; ;
0x000810C 205 [F0] not R5
0x0008110 205 mov R4, 0
0x0008112 205 ld R3, (R2) ; ;
0x0008114 206 jmp R8 ; ;
PC = 0x00008114
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002004
R3 = 0x00000008
R4 = 0x10000000
R5 = 0x0000ffff
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00008100
SP = 0x00000ff0
M1
> s
> s
> s
> s / c
>
W3
Pipline Cycles : 0104
Instruction Steps : 0208
W4
Stage IF DC EX WB
Pc 0x0008100 - 0x0008114 0x000810C
Slot X mov jmp not
Slot Y mov mov
Slot Z ld
M3
W7

FIG. 33


W1
W5
PC LINE FLG MNEMONIC
0x0008100 200 mov R0, 0
0x0008102 201 mov R1, 1 ; ;
0x0008104 202 add R0, R1 ; ;
0x0008106 203 ld R3, (R2+)
0x0008108 204 [F1] or R4, R5 ; ;
0x000810A 204 cmp F0, R0, R3 ; ;
0x000810C 205 [F0] not R5
0x0008110 205 mov R4, 0
0x0008112 205 ld R3, (R2) ; ;
0x0008114 206 jmp R8 ; ;
PC = 0x00008114
R0 = 0x00000001
R1 = 0x00000001
R2 = 0x00002004
R3 = 0x00000008
R4 = 0x10000000
R5 = 0x0000ffff
R6 = 0x00000000
R7 = 0x00000000
R8 = 0x00008100
SP = 0x00000ff0
M1
> s
> s
> s / c
> s / c
>
W3
Pipline Cycles : 0105
Instruction Steps : 0208
W4
Stage IF DC EX WB
Pc 0x0008104 0x0008100 - 0x0008114
Slot X add mov jmp
Slot Y ld mov
Slot Z or
W7